ABSTRACT OF THE DISCLOSURE

A parallel processing system and method for performing processing tasks in parallel on a plurality of processors breaks down a processing task into a plurality of self-contained task objects, each of which has one or more "data-waiting" slots for receiving a respective data input required for performing a computational step. The task objects are maintained in a "waiting" state while awaiting one or more inputs to fill its slots. When all slots are filled, the task object is placed in an "active" state and can be performed on a processor without waiting for any other input. The "active" tasks objects are placed in a queue and assigned to a next available processor. The status of the task object is changed to a "dead" state when the computation has been completed, and dead task objects are removed from memory at periodic intervals. This method is well suited to computer graphics (CG) rendering, and particularly to the shading task which can be broken up into task spaces of task objects for shading each pixel of an image frame based upon light sources in the scene. By allowing shading task objects to be defined with "data-waiting" slots for light/color data input, and placing task objects in the "active" state for processing by the next available one of an array of processors, the parallel processing efficiency can be kept high without wasting processing resources waiting for return of data.